

# NoSql en sistemas distribuidos sobre Cluster Hadoop

Adriana E. Martin<sup>1</sup>, Susana B. Chavez<sup>2</sup>, Nelson R. Rodríguez<sup>3</sup>, María A. Murazzo<sup>4</sup>

Departamento e Instituto de Informática - F.C.E.F. y N. - U.N.S.J.  
Complejo Universitario Islas Malvinas, Av. I. de la Roza 590 (O), CP: 5402 Rivadavia,  
San Juan. Tel:0264 4234129

<sup>1</sup>arianamartinsj@gmail.com <sup>2</sup>schavez@iinfo.unsj.edu.ar <sup>3</sup>nelson@iinfo.unsj.edu.ar  
<sup>4</sup>marite@unsj-cuim.edu.ar

## Resumen

Con el gran avance de los sistemas distribuidos en la web y las tecnologías informáticas distribuidas como Cluster y Cloud, el software como servicio (SaaS), los servicios en el Cloud y los constantes requerimientos de procesamiento y análisis a gran escala de enormes cantidades de datos, los sistemas tradicionales de base de datos son insuficientes. Las Bases de Datos NoSQL llenan una importante carencia de las bases de datos relacionales en cuanto a la capacidad que estas tienen en escalabilidad, distribución y manejo de datos no estructurados. Estas 3 características son cada día más relevantes debido precisamente al avance de Cloud Computing, y a los múltiples y diversos servicios cuyo crecimiento y replicación distribuida son extremadamente necesarios.

El framework Hadoop Map/Reduce permiten realizar procesamiento distribuido de grandes volúmenes de datos. La naturaleza distribuida de Cassandra coincide muy bien con la naturaleza distribuida de MapReduce, para ejecutar consultas sobre datos que abarcan múltiples nodos. La sencillez y flexibilidad de Cassandra, su lenguaje

de consulta (CQL) y el soporte del controlador múltiple como la capa de almacenamiento de datos para MapReduce, tanto para la entrada y salida de datos; permiten sobreponerse a las limitaciones del Sistema de Archivo de Hadoop (HDFS)

**Palabras Claves:** Computación distribuida, Cloud Computing, Nosql, MapReduce

## Contexto

El presente trabajo se encuadra dentro del área *Base de Datos y Minería de Datos*, y se enmarca dentro del proyecto de investigación: **Evaluación de arquitecturas distribuidas de commodity basadas en software libre**, el cual se ha presentado en una nueva convocatoria, y cuyo resultados se espera que estén para abril del presente año. Tendrá como unidades ejecutoras al Departamento e Instituto de Informática de la FCEFN de la UNSJ.

## Introducción

La computación distribuida es un modelo destinado a resolver problemas

de cómputo masivo utilizando un gran número de computadoras organizadas sobre una infraestructura de telecomunicaciones distribuida. El objetivo principal de la computación distribuida es contar con un entorno de entidades distribuidas que cooperen para resolver un problema que no puede ser resuelto individualmente. La variedad de estos sistemas pueden incluir servidores de aplicaciones, cloud privados, pequeños centros de datos, cluster de cómputo y cluster para almacenamiento y búsqueda de datos. También es posible la construcción de arquitecturas híbridas que permitan montar cluster como servicio (CaaS), facilitando hacer el mejor aprovechamiento de los recursos virtualizados en el cloud, con el objeto de utilizar ecosistemas, tales como: Hadoop MapReduce para realizar un procesamiento distribuido de grandes volúmenes de datos.

Con el fin de gestionar estas grandes cantidades de datos, que no tienen en general una estructura determinada, surgen las Bases de Datos NoSQL[1].

Apache Cassandra es una base de datos NoSQL distribuida y de código abierto, cuya principal característica es que fusiona Dynamo, de Amazon con BigTabla, de Google, siendo ambas implementaciones de código cerrado. Está basada en un modelo de almacenamiento de «clave-valor», escrita en Java. Permite almacenar grandes volúmenes de datos en forma distribuida. Su objetivo principal es la escalabilidad lineal y la disponibilidad. [2].

La arquitectura distribuida de Cassandra está basada en una serie de nodos iguales que se comunican con un protocolo P2P (Peer-to-peer). En este protocolo, la Red está descentralizada, no tiene clientes ni servidores fijos, sino

que tiene una serie de nodos que se comportan simultáneamente como clientes y servidores de los demás nodos de la red. Cada nodo puede iniciar, detener o completar una transacción compatible. Contrasta con el modelo cliente-servidor. Esto lleva a una comunicación en la que la redundancia es máxima.

En las versiones iniciales utilizaba un API propia para poder acceder a la base de datos. En los últimos tiempos están apostando por un lenguaje denominado CQL (Cassandra Query Language) que posee una sintaxis similar a SQL aunque con muchas menos funcionalidades. Esto hace que iniciarse en el uso de la misma sea más sencillo. Permite acceder en Java desde JDBC.

“Apache Cassandra se ha convertido en una de las bases de datos NoSQL más utilizados del mundo y sirve como columna vertebral de algunas aplicaciones muy populares hoy en día”, Por ejemplo, lo usa Twitter para su plataforma. Su objetivo principal es la escalabilidad lineal y la disponibilidad. Permite el uso de Hadoop para implementar Map-Reduce, ya que Hadoop puede trabajar directamente con cualquier sistema de archivos distribuido [3]

Está disponible en Linux rpm, Deb, y también existe, un instalador MSI de Windows, y Mac OS X binary.

MapReduce es fundamental en las bases de datos NoSQL para permitir la utilización de funciones de agregación de datos, ya que al carecer de esquema son mucho más complicadas que en las bases de datos relacionales clásicas RDBMS (Relational Database Management System).

MapReduce es un paradigma de programación para el procesamiento y manejo de grandes conjuntos de datos. Se divide las solicitudes en solicitudes más pequeñas que se envían a muchos servidores pequeños para ser procesados de una manera paralela. Como resultado, puede procesar grandes conjuntos de datos muy rápidamente. Común / Core es un paquete que contiene las bibliotecas y los servicios públicos de apoyo a los módulos de Hadoop. El hilo es una plataforma de gestión de recursos incluidos para la gestión de los recursos de computación y programación de tareas de Hadoop.

Hadoop es un framework de Apache de código abierto basado en Java, se ejecuta sobre la máquina virtual de Java (necesario JRE 1.6 o superior) para procesar y consultar grandes volúmenes de datos en grandes clusters de hardware commodity.

La arquitectura Hadoop esta formada por los siguientes módulos:

1. Hadoop Common: Utilidades comunes en las que se apoyan los otros módulos. Proporciona abstracciones a nivel de sistema de archivos o sistema operativo. Contiene los archivos .jar (Java ARchive) y scripts necesarios para iniciar Hadoop.

– Hadoop Distributed File System (HDFS): Sistema de archivos distribuido, que almacena datos en las máquinas commodity, que proporciona acceso de alto rendimiento a datos.

– Hadoop YARN: Plataforma de administración responsable de la gestión de los recursos en el cluster y de los medios para el desarrollo de aplicaciones de los usuarios.

– Hadoop MapReduce: Modelo de programación para el procesamiento de grandes volúmenes de datos.[4]

Hadoop utiliza una arquitectura escalable que hace uso de servidores básicos configurados como un conjunto, donde cada servidor posee unidades de disco internas de bajo costo. Los datos de Hadoop se dividen en bloques y se extienden a lo largo de un clúster. Una vez que esto sucede, las tareas de MapReduce pueden llevarse a cabo en los subconjuntos más pequeños de datos que pueden hacer que un muy gran conjunto de datos en general, cumpliendo así el tipo de escalabilidad necesaria para el procesamiento de grandes volúmenes de datos.[5]

Los cluster Hadoop son conocidos por aumentar la velocidad de las aplicaciones de análisis de datos. También son altamente escalable: Si la potencia de procesamiento está desbordado por volúmenes de datos cada vez mayor, nodos adicionales del clúster se pueden agregar para aumentar el rendimiento. Los cluster Hadoop también son altamente resistentes a fallos, ya que cada pieza de datos se copia en otros nodos del clúster, lo que asegura que los datos no se pierden si falla un nodo.

Un clúster Hadoop esta formado de la siguiente manera:

- Un nodo maestro. En el nodo maestro encontramos un JobTracker (rastreador de trabajos), un TaskTracker (rastreador de tareas), un NameNode (nodo de nombres) y un DataNode (nodo de datos).

– varios nodos esclavos. Un nodo esclavo actúa tanto de DataNode como de TaskTracker normalmente.

Hadoop Distributed File System (HDFS) es un sistema de archivos distribuido que se parece a cualquier otro sistema de archivos, excepto que cuando se mueve un archivo en HDFS, este archivo se divide en muchos archivos pequeños, cada uno de esos archivos se replica y se almacena en (por lo general, puede personalizar) 3 servidores para las limitaciones de tolerancia a fallos.

En general, esta estrategia de divide y vencerás de procesamiento de datos no es nada nuevo, pero la combinación de HDFS, siendo el software de código abierto (que supera la necesidad de soluciones de almacenamiento especializados de alto precio), y su capacidad para llevar a cabo un cierto grado de redundancia y tolerancia a fallos automático, hacen que sea popular para las empresas modernas que buscan soluciones de análisis de proceso por lotes

### **Líneas de Investigación y desarrollo**

Las diferencias entre los sistemas distribuidos es que plantean numerosos desafíos a resolver como son la configuración adecuada, optimización de los recursos, balance de carga, uso eficiente de la red, distribución adecuada de los datos, configuración de frameworks, elección del tipo de hipervisor, uso y configuración del mismo, opciones de virtualización, etc. El objetivo del presente proyecto es evaluar las distintas variantes de sistemas distribuidos de recursos limitados, y proponer en función de los resultados las opciones más convenientes para cada caso. Vale aclarar, que cuando se decide contar con una solución basada en cloud es de suma importancia analizar cual despliegue se usará, pues esta elección impacta directamente en la performance

de la solución. Hay que recordar que cualquier elección que implique el uso intensivo de comunicaciones degrada el desempeño por los retardos incurridos en el proceso de comunicación en si [6] [7].

También es importante destacar que es posible la construcción de arquitecturas híbridas que permitan montar cluster como servicio (CaaS), para lograr el mejor aprovechamiento de los recursos en el cloud, con el objeto de utilizar Hadoop MapReduce para realizar un procesamiento distribuido de grandes volúmenes de datos

### **Resultados Obtenidos**

A pesar de que esta línea de investigación recién ha sido presentada para los años 2016 y 2017. Durante los últimos cinco años se trabajó en proyectos sobre Cloud Computing y en particular durante los últimos dos años sobre Cloud híbridos. La experiencia sobre los Cloud privados, junto con líneas de investigación anteriores [7], impulsó esta línea de investigación. El grupo ha realizado nueve publicaciones en el área durante el último año: tres trabajos en el WICC 2015, un trabajo en el CACIC 2015, dos trabajos en las Jornadas de Cloud Computing, además se realizaron tres publicaciones en revistas científicas.

Se han aprobado tres tesinas de grado y un trabajo de especialización.

### **Objetivo**

En este proyecto se enfocaran las investigaciones en los sistemas de cómputo distribuidos, los cuales permiten realizar de manera más eficiente tareas de computación de alta prestaciones basadas en el paradigma de memoria distribuida. Ejemplos de Arquitecturas que soportan este tipo de

sistemas distribuidos son los cluster y el cloud computing.

En particular este trabajo tiene como objetivo instalar una base de datos Nosql, en particular Cassandra, sobre un cluster montado como servicio (CaaS), para luego utilizar el framework Hadoop para realizar procesos distribuidos de grandes volúmenes de datos.

### **Formación de Recursos humanos**

El equipo de trabajo está compuesto por los seis docentes-investigadores que figuran en este trabajo y cuatro alumnos.

Se están realizando cuatro tesinas de licenciatura una sobre evaluación de algoritmos de algebra lineal sobre arquitecturas diversas, otra sobre Cloud Computing Privado, otra sobre dispositivos de juegos aplicados a salud y otra sobre SOA aplicada a Cloud. Se espera realizar también una tesis de maestría sobre Metodologías de desarrollo aplicadas a SaaS, otra sobre bases de datos NoSQL y otra sobre algoritmos de Cómputo Intensivo para Big Data y su implementación en Clouds. Además aumentar el número de publicaciones. Por otro lado también se prevé la divulgación de varios temas investigados por medio de cursos de postgrado y actualización o publicaciones de divulgación.

### **Referencias**

- [1] Martin, A., Chavez, S., Murazzo, M., Rodríguez, N., Valenzuela, A. “MongoDB en ambiente Cloud Híbrido con OpenStack”. Wicc2015.
- [2] <http://cassandra.apache.org>
- [3] [www.siliconweeks.com](http://www.siliconweeks.com)
- [4] <https://www.mongodb.com/hadoop-and-mongodb>
- [5] <http://hadoop.apache.org/>
- [6] Murazzo, M., Rodríguez, N., Medel, D., Chavez, S., Martin, A., Valenzuela, F. “Análisis de mejora en la escalabilidad de las infraestructuras de cloud computing”. III Jornadas de Cloud Computing. UNLP (2015).
- [7] Murazzo, M., Tinetti, F., Rodríguez, N. “Despliegue de una Infraestructura Cloud Privada de Código Abierto”. III Jornadas de Cloud Computing UNLP (2015).